RL-TR-97-66
Final Technical Report
August 1997

ANALYSIS OF MOUNTAINTOP DATA REDUCTIONS

Decision-Science Applications, Inc.

Sponsored by
Advanced Research Projects Agency
ARPA Order No. A0241

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-97-66 has been reviewed and is approved for publication.

APPROVED:

E. DOUGLAS LYNCH
Project Engineer

FOR THE DIRECTOR:

GARY D. BARMORE, Major, USAF
Deputy Director Surveillance & Photonics

ANALYSIS OF MOUNTAINTOP DATA REDUCTIONS

Contractor: Decision-Science Applications, Inc.
Contract Number: F30602-93-D-0081, 008
Effective Date of Contract: 24 May 95
Contract Expiration Date: 23 May 96
Program Code Number: SC1030
Short Title of Work: Analysis of Mountaintop Data Reductions
Period of Work Covered: May 95 - May 96

Principal Investigator: Lisa Slaski
Phone: (315) 339-6184
RL Project Engineer: E. Douglas Lynch
Phone: (315) 330-4515

Approved for public release; distribution unlimited.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by E. Douglas Lynch, RL/OCSA, 26 Electronic Pky, Rome, NY.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1997	Final May 95 - May 96

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
ANALYSIS OF MOUNTAINTOP DATA REDUCTIONS	C - F30602-93-D-0081, 008 PE - 63226E PR - MSEF TA - QF WU - 01
6. AUTHOR(S) Lisa Slaski	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Decision-Science Applications 1300 Floyd Avenue, Building B Rome NY 13440	8. PERFORMING ORGANIZATION REPORT NUMBER N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Advanced Research Projects Agency, 3701 North Fairfax Drive, Arlington VA 22203-1714 Rome Laboratory/OCSA, 26 Electronic Pky, Rome NY 13441-4514	10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-97-66

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: E. Douglas Lynch/OCSA/(315) 330-4515

12a. DISTRIBUTION AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution unlimited.	

13. ABSTRACT *(Maximum 200 words)*

The Ozturk Algorithm is a statistical algorithm which determine the best probability density function (PDF) to approximate the input data. This report describes in detail the research performed, the insights gained and the results obtained during this effort. The work performed was to provide all the necessary items to automate the application of the Ozturk Algorithm to radar clutter data under the Rome Laboratory Space-Time Adaptive Processing/Algorithm Development Tool (RLSTAP/ADT). These items are divided into five sections for this report; Requirements for Measured Clutter Data, Khoros Flow Control Mechanism, Modification of the Ozturk Algorithm Modules, Implementation of the Graphics for a Range-Angle Map, and Implementation of the Ozturk Algorithm Loop Application. The automated range-angle Ozturk Algorithm loop application was completed under RLSTAP/ADT and all necessary modules and change sets were provided for integration into Rome Lab's resident version.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Ozturk Algorithm, PDF, RLSTAP/ADT, Measured Clutter Data, Khoros Flow Control Mechanism, Range Angle Map	44
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

Section/Title **Page**

TABLE OF FIGURES

Figure/Title **Page**

1. INTRODUCTION

The major thrust of the work performed over the course of this effort was to provide all the necessary items to automate the application of the Ozturk Algorithm to radar clutter data under RLSTAP/ADT.

The Ozturk Algorithm is a statistical algorithm which determines the best probability density function (PDF) to approximate the input data. Furthermore the algorithm provides an estimate of the PDF's location, scale and shape parameters. The advantage of the Ozturk Algorithm is that it works with small data set sizes, on the order of 100 samples, and selects a PDF from a variety of PDF's. Whereas classical techniques require thousands of samples and only test for one PDF at a time. Details of the Ozturk Algorithm are provided in References [2] through [6].

The desired application of the Ozturk Algorithm is to perform a statistical analysis of a range-angle-frequency (RAF) sector, where the sector will be allowed to slide in both range and angle to the full dimensions of the clutter data array. For each sector location in range and angle the Ozturk statistics will be stored. At the end of the RAF loop the statistics will then be plotted in a range-angle map style. Details of this type of application for the Ozturk Algorithm are provided in Reference [1].

The RAF loop application could be accomplished by a single module or lineup that provided the RAF loop internally. However, this would restrict all future statistical sector analysis to the Ozturk Algorithm, without extensive software development. To be more flexible it is desired to implement the RAF loop using the Khoros flow control modules. This then provides both a loop example for the user and programmer, as well as providing a simple method to incorporate other statistical analysis modules within RLSTAP for sector analysis.

To perform this task it was necessary to develop modules that 1) read in the RSTER radar range-angle clutter data, 2) use the Khoros flow control modules to perform a looping mechanism, 3) define the desired data sector and extract the data to be processed for each instance of the loop, 4) extract the desired resultant Ozturk statistic and display the result in a range-angle map format. Furthermore, it was necessary to extensively modify previously developed Ozturk Algorithm modules in order that they could be used within the Khoros loop mechanism.

Originally, it was assumed that the Khoros graphics routine, XPRISM, could be used to provide the necessary range-angle map display. However, during the course of this effort, it was found that XPRISM was insufficient, requiring the development of new graphics routines based on the Xlib and Athena widget set libraries. These public domain graphics libraries provided the necessary capabilities without requiring researchers to purchase a user-license.

1.1 MAJOR ACCOMPLISHMENTS

The automated range-angle Ozturk Algorithm loop application was completed under RLSTAP and all necessary modules and change sets were provided for integration into RL's resident version. This effort included the following major accomplishments:

1. Developed a module to provide a general plot utility for a range-angle map without the dependence on PVWave or any other commercial graphics package. Module: RAMap.

2. Developed modules to provide the XY Terrain Cover and Height maps without the dependence on PVWave or any other commercial graphics package. Module: PltXYMap.

3. Developed modules to read in, compute and plot measured clutter data from the RSTER radar. Modules: RSTERcnr, RSTERbsc, RSTERmsh, and PltRAMap.

4. Investigated the use of the Khoros flow control modules which were previously incorporated into RLSTAP. Provided a memo detailing the conceptual use of the two available loop types.

5. Validated and extensively modified the Ozturk Algorithm modules so that they might be used within a Khoros flow control loop. Modules: OzID, OzGen, OzUV, OzPest, OzLoc, OzGraph, and OzGOF.

6. Developed modules to provide automation of the range-angle-frequency loop application desired for the automated Ozturk Algorithm loop application. Modules: RafSec, RafExt, and RafOzPlt.

The following sections of this report describe in detail the research performed, the insights gained and the results obtained during this effort.

2. REQUIREMENTS FOR MEASURED CLUTTER DATA

In order to apply the Ozturk Algorithm on measured clutter data, it is first necessary to make this data available within RLSTAP/ADT. To do this, several modules were developed.

2.1 EXISTING MEASURED CLUTTER DATA SETS

The existing measured clutter data set used for the development of the required modules was taken with the RSTER radar at the WSMR/NOP site on 27 Aug. 93 by Lincoln Lab. The uncalibrated data set name is "Multiple Frequency Monostatic Clutter Map Data Tape, Fc = 432, 435, 438, 485". This data set consists of five overlapping range rings of data for each of four frequencies (432, 435, 438, and 485 MHz). Each frequency set contains an acal, rcal and ncal file. This data set is uncalibrated data only.

For the WSMR site, to obtain the full operational range of the radar, only the data from the first three range rings are required. As each frequency data set and each range ring are separate experiment collections, each set will start at a different azimuth and as stated, each range ring overlaps the previous. Furthermore, due to wind stress, etc., the data's azimuth values do not increment uniformly. Thus, to obtain a full data set, the data from each range ring must be read in separately, then sampled and concatenated appropriately to obtain a single uniformly sampled range-angle map.

For a given range ring, each CPI of data represents a different azimuth value. Thus, to develop a CNR map for one range ring, it is necessary to read in each CPI output from the module RSTERin, perform a beamform across the antenna channels, divide by the noise estimate and save into a range-angle data structure. This is performed by the module RSTERcnr.

Furthermore, to obtain the noise estimate the ncal file must be read in, and for each CPI the data must be beamformed and the average dB value determined. Presently, this can be performed by using the RSTERcnr module with the noise estimate parameter in the control pane set to 0.00 dB, and feeding the resultant output of RSTERcnr into PlotLine. Then through the Plot Info button within PlotLine, the average value may be obtained. This value must then be used as the noise estimate for the CNR computation.

A more automated method of determining the noise estimate has not been provided as yet, since the data files to be provided to the general user will be calibrated, not uncalibrated data. At this point it is unknown to the researcher, if the noise estimate will be part of the Matlab header file or still be contained in a separate file. In either case, the modification of this module should be performed at some point in the future to accommodate the implementation of the noise estimate for a calibrated data set.

2.2 APPLICATION OF THE OZTURK ALGORITHM ON MEASURED CLUTTER DATA

The purpose of this effort was to automate the application of the Ozturk Algorithm on radar clutter data. The specific conceptual application which has been duplicated in this effort, has been fully detailed in Reference [1]. However, a few of the basic concepts must be outlined here to understand the development of the associated modules. Also, note that the original work outlined in Reference [1] was performed over several days, "manually" extracting the desired data vectors within a Matlab code environment and handwriting the results of the Ozturk Algorithm in tabular form.

The Ozturk Algorithm expects that its input data will be identically independent data. However, the best one can insure from measured clutter data is that the data is uncorrelated. To help insure this, it was determined that the dependency on the range of each clutter data point back to the radar site must be eliminated as much as possible. Thus, the Backscatter Coefficient of the clutter data is computed, with a flat earth model as presented in Reference [7]. This is performed by the module RSTERbsc.

A better method for obtaining the Backscatter Coefficient of the clutter data, but not implemented here or previously in the Matlab environment, would be to incorporate the USGS terrain elevation information into the computation to determine the exact distance to the radar site for each range bin. This would provide more accurate results, but it would also be an extensive development effort.

To provide Ozturk statistics for a range-angle map, a desired range-angle data sector must be defined for the extraction of the desired clutter data points. This fixed size data sector may then be moved in range and angle and the Ozturk statistics plotted at the center of each range-angle data sector location.

Again to help insure uncorrelated data, the desired sampling of the clutter data must be greater than or equal to every range resolution cell and every azimuth beamwidth. Also, in order to reduce the overall physical size of the range-angle data sector to be analyzed, samples from two or more frequencies can be used. This fixed size data sector is defined in module RafSec.

2.3 ACQUIRING RAF CLUTTER DATA FOR PROCESSING WITHIN RLSTAP

Since each range ring and each frequency data set start at different arbitrary angles, it is necessary to sample the data from a consistent starting angle, whenever two or more data files are desired to be concatenated. Once the CNR or Backscatter Coefficient data has been obtained for each desired frequency and range ring, then a single range-angle-frequency data structure may be obtained through the use of RSTERmsh. The module RSTERmsh takes up to three range rings of a given frequency and samples the data in uniform angle increment values, from a specified starting angle, and extracts and concatenates the appropriate range values from each range ring, to produce a single uniformly sampled range-angle data array for each frequency. Then the

different frequency data arrays may simply be concatenated by bands to obtain the range-angle-frequency data cube to be used within the Ozturk Algorithm loop application.

Figure 2-1 provides an example of a complete lineup for a two frequency, single range ring data array. Even though only one range ring is desired it is best to execute the RSTERmsh module to obtain uniform azimuth sampling of the data array for downstream processing. Note that RSTERin, RSTERcnr (with a 0.0dB noise estimate) and PlotLine must be executed for the ncal file associated with each frequency to obtain the average noise estimate from the Plot Info button of PlotLine. This is shown in the top and third lines in the lineup of Figure 2-1. Next, using the average value from the ncal file as the noise estimate, RSTERin, RSTERcnr, RSTERbsc, and RSTERmsh may be executed for each desired frequency and Concat is used to concatenate the data in the frequency dimension, or bands, to obtain the desired resultant data set. This is shown in the second and fourth rows of the lineup.

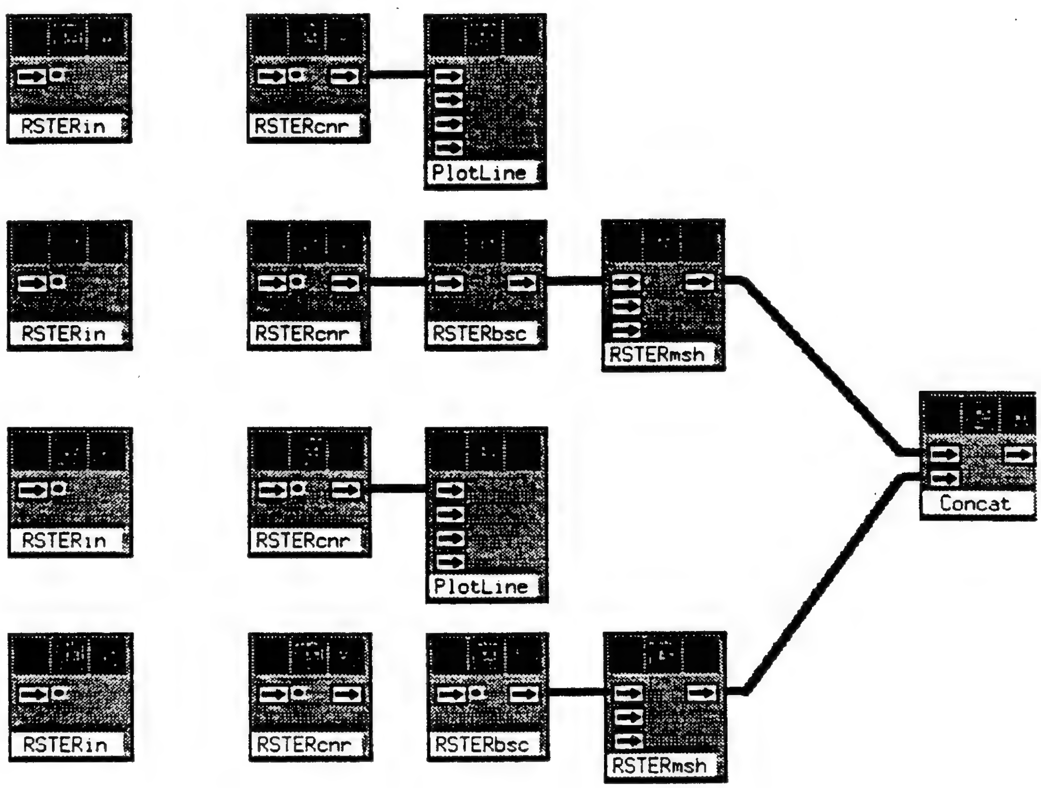

Figure 2-1. Two Frequency, Single Range Ring, Measured Backscatter Coefficient, Module Lineup

For a two range ring data set, simply add a similar lineup of RSTERin, RSTERcnr and RSTERbsc for the second range ring, with the result provided as the second input to RSTERmsh for each frequency. Note that the noise estimate for each frequency is used for all range rings for that frequency.

3. KHOROS FLOW CONTROL MECHANISM

Under the original Khoros/Cantata framework and incorporated into RLSTAP/ADT, there exists several flow control modules which are intended to be used to control the flow of a lineup through the use of count loops, if/else statements, etc. These modules can be found under the push button "Utilities" and its menu item "Flow Control", found in the upper left area of the cantata form.

The flow control module selections include:

Count-Loop: module name: count_loop

While-Loop: module name: while_loop

If-Then-Else; module name: if_else

Merge; module name: merge

Trigger; module name: trigger

Expression: module name: expr

Switch module name: switch

A detailed description of two simplistic Count-Loop flow control examples were provided in a memo written by the author during the course of this contract (Reference 8).

3.1 BASIC MODULE REQUIREMENTS

In order to produce a loop application, the modules which the user wants to control within the loop must satisfy the following conditions:

1. All modules to be controlled within the loop, including the first module, must have an input file specifier so that the count_loop module can be visually connected to the modules within the loop. This connection also allows the count_loop to keep track of the loop and allows its parameters to be used as arguments within the panes of the other modules within the loop.

2. Any specific module parameter that is desired to be controlled by the loop, must exist as an input parameter in the pane of the appropriate module(s) within the loop.

3. Any desired output from the loop must exist within the VIFF data file (or header).

If one or more of these conditions does not exist, then existing modules must be modified or new modules created to handle the desired application.

Also, it should be noted that, in general, all modules to be controlled within the loop should not have any interactive user input, or spawn another process, such as

graphics provided by XPRISM, as this apparently is not always handled well within the event loop by the Khoros flow control modules and may cause problems.

3.2 ADVANTAGES/DISADVANTAGES OF THE LOOP APPLICATION

The flow control modules within RLSTAP provide a powerful tool for the user to create custom lineups beyond anything that RL might envision. The obvious advantage is the flexibility it provides to the user for producing custom applications.

However, there are disadvantages to using the loop methods. The most important being that creating a loop application creates a large overhead which effects computational efficiency. Thus, this mechanism works best for small numbers of iterations and for custom, one time applications.

If the user is expected to use the particular loop algorithm many times, it is best to create a lineup or single module that makes sense for the algorithm, internalizing the loop, rather than relying on the flow control modules to create the algorithm.

Another disadvantage is that a given loop application may not work exactly as expected due to the original intent and design of the modules used within the loop, as noted in Reference [8].

Also, at present, within the loop, RLSTAP/ADT adds each module's header information to the VIFF header. This header can become quite large, to the point of significantly slowing down any module that requires information to be read in from the VIFF header. This is a disadvantage of the Khoros flow control mechanism as currently implemented within RLSTAP. However, an SPR has been issued to address this problem.

3.3 THE FLOW CONTROL MODULES USED IN THE OZTURK LOOP APPLICATION

The automated Ozturk Algorithm loop application uses the flow control modules to set up a loop for which, each instance of the loop, the fixed range-angle data sector is "moved" to a new range-angle location, the desired data is extracted from the sector and the Ozturk Algorithm is performed on this data. At the end of the loop the resultant Ozturk statistic is plotted for each value of range and angle in a range-angle map format.

3.3.1 <u>The Count Loop Module</u>

This module sets up a series of parameters to control a count loop. These parameters, which can be expressions, are passed within the loop. To use the loop parameter as an input argument to another module, place a "$" in front of the label that was setup in the count_loop module, as the input within the control pane for the desired module.

There are two types of loops that can be set up. One type uses feedback of the data as the loop input, and the other does not provide feedback of the data (the loop input remains constant). These two types of loops are illustrated by Figures 3-1 and 3-2, each of which perform a similar simple algorithm. They both concatenate a random data vector generated by OzGen to itself.

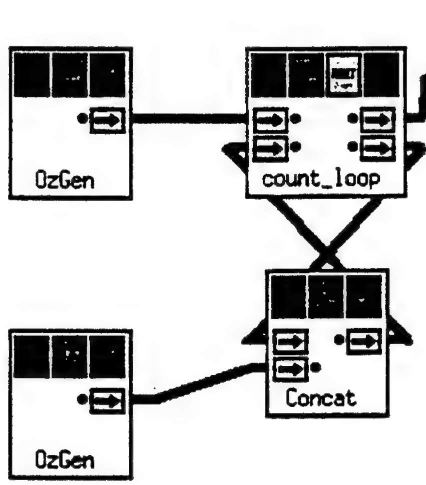

Figure 3-1. Simple-Feedback Loop Application

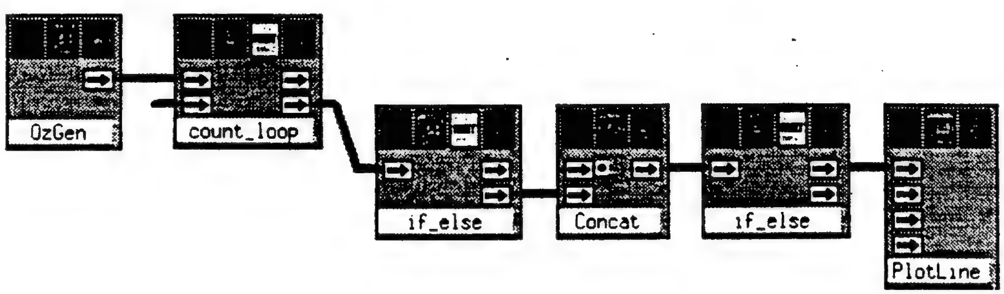

Figure 3-2. Simple No-Feedback Loop Application

As the loop is executed, the count_loop module keeps track of the loop parameter, initializing it on the first loop and incrementing it until the desired final value is reached. The first time the loop is executed the count_loop passes the input data (top left input) to the loop output (bottom right output). Thereafter the loop input data (bottom left input) is used until the last loop, when it is passed to the output (top right output). When attempting to setup or understand a loop application, it is important to note that the data filenames within a loop remain constant.

In the feedback loop example of Figure 3-1, there is no specific setup necessary. Simply connecting these modules in this manner will provide the desired output. If both

OzGen modules are setup to provide the same data, and as the first loop performs a concatenation of the data, the data will be concatenated N times, for N loop iterations.

In the "no-feedback" loop example of Figure 3-2, two if else modules are used to determine the occurrence of the first and last loops. This module type is described in section 3.3.3. During the first loop, the first if_else module will store the data at this point into a file, say "test". On subsequent loops the Concat module is set to read in the file "test", concatenate the current loop data to it, and also store it back into "test". The second if_else only executes PlotLine when the last loop occurs. Thus, the data in this case will be concatenated N-1 times, for N loop iterations.

The automated Ozturk Algorithm loop application uses the no-feedback loop control, as the desired data input to the loop is constant. The desired data is the full range-angle clutter map input, from which the desired data sector is extracted for each instance of the loop. The automated Ozturk Algorithm loop application is described in detail in section 6.

3.3.2 The Expression Module

This module simply sets up an expression for a given variable. The variable may then be used as an input argument to another module by placing the variable name with a "$" in front of it, in the control pane of the desired module.

For the automated Ozturk Algorithm loop application, this module is used to provide desired arguments to both the Count_Loop modules for the range-angle loop control and the range-angle-frequency extraction routine, RafExt.

3.3.3 The If-Then-Else Module

This module simply provides a desired If-Then-Else statement. The input arguments of this module may be expressions. Typically, the expressions take on the syntax of c-code.

For the automated Ozturk Algorithm loop application, this module is used to determine the first instance of the range-angle loop and store the resultant data into the specified output VIFF file. After the first instance the resultant data is then concatenated by bands to the previous data. It is also used to determine the last instance of the loop, at which time the following modules provide a range-angle type plot of the desired resultant Ozturk statistic.

4. MODIFICATION OF THE OZTURK ALGORITHM MODULES

Previous to the start of this effort, a basic set of modules had been developed for applying the Ozturk Algorithm to a sample data set. However, these modules were not developed with the intent to use them within a Khoros flow control mechanism. The original output of these modules was generally an ASCII file.

In order to apply the Ozturk Algorithm within a Khoros loop, the associated modules were extensively modified to extract the graphical results into separate modules and provide the passing of VIFF files. Also, the algorithm was somewhat reorganized into smaller application modules. Figure 4-1 provides a typical lineup for applying the Ozturk Algorithm to a sample data set.

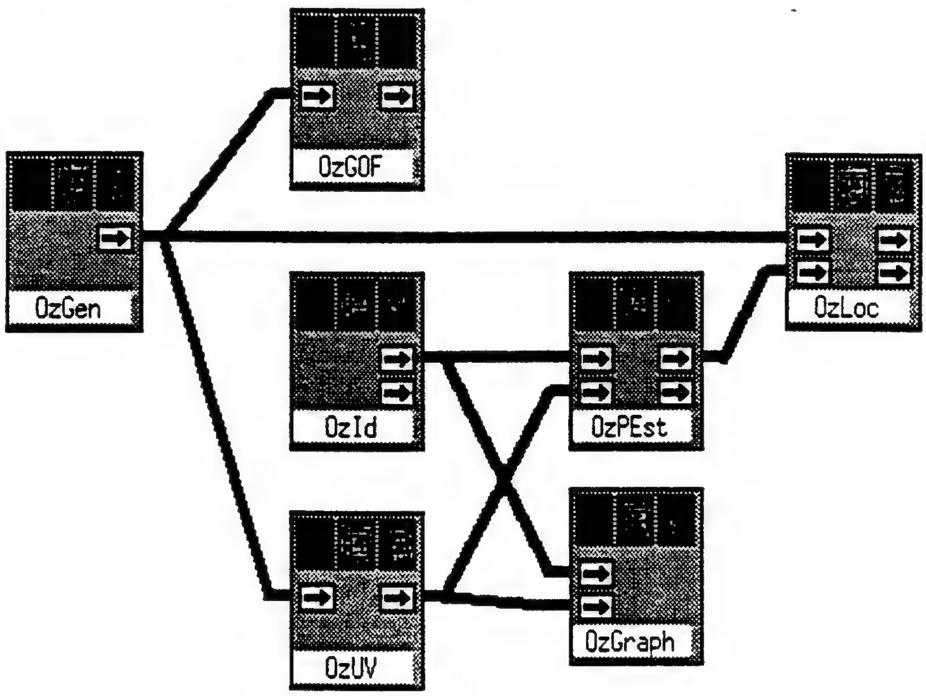

Figure 4-1. Simple Ozturk Application Module Lineup

The OzGen module simply provides a 3D array of a total of N random variates from the distributions available within the Ozturk Approximation Chart.

The OzID module provides the Ozturk Approximation Chart, which is valid only for the number of samples, N, for which it was created. Since this portion of the algorithm takes the most time, it is typically generated and stored for each desired value of N.

The OzGOF module provides the graphics for the Goodness-of-Fit Test from the sample data for a specified Null Hypothesis.

The OzUV module determines the sample data linked vector, where the last point is the Ozturk statistic. The output data is an IQ array of U/V data points defining the linked vector.

The OzGraph module provides the Ozturk Approximation (or ID) Chart with the sample data linked vector overlaid.

The OzPest module determines the PDF shape parameter estimation of either all 16 PDFs and rank orders them from the most to least likely fit, or estimates the shape parameters for only 1 PDF as selected by the user. It requires an Ozturk Approximation (or ID) Chart to perform this computation. These charts are generated by OzID and are only valid for the sample size for which they were developed.

The OzPest output data format is 8 Ozturk statistics (rows), by either 1 or 16 PDFs (columns) which are of implied rank order.

The 8 Ozturk statistics are:

1. U Co-ordinate
2. V Co-ordinate
3. Distribution #
4. Distance
5. Location parameter
6. Scale parameter
7. First shape parameter
8. Second shape parameter

The location and scale parameter estimates are determined in the OzLoc module, which requires the original sample data as well as the shape parameter estimate for the computation. The output data is of the same format as that for the OzPest module.

Other statistical tools such as the Histogram and PDF Envelope modules may be used to check the validity of the approximation performed by the Ozturk Algorithm. However, not all of the available PDF's within the Ozturk Algorithm are available within the PDF Envelope module.

5. IMPLEMENTATION OF THE GRAPHICS FOR A RANGE-ANGLE MAP

Within the physical model simulation of RLSTAP there are ten range-angle map style plots required to provide scene plan information to the user. These plots include:

1. RA Terrain Cover
2. RA Terrain Height
3. Clutter-to-Noise (CNR) Map
4. Backscatter Coefficient Map
5. Line-of-Sight (LOS) Map
6. Clutter Intensity Map
7. Doppler Map
8. Backscatter Map
9. Grazing Angle Map
10. Scene Plan Map

In addition, the measured data CNR and Backscatter Coefficient maps and the Ozturk statistics from the loop application also need to be plotted in a range-angle map style. Also, the two XY Terrain Cover and Terrain Height maps from the physical model simulation similarly need many of the desired features of a range-angle map routine.

At the start of this effort, the only method for plotting a range-angle map was through the use of the commercial package PVWAVE. As many potential users of RLSTAP will not have this commercial software available to them, it was necessary to provide a similar capability using software directly available within RLSTAP. It was originally thought that the Khoros plotting module XPRISM would be able to provide this capability.

5.1 INVESTIGATION OF XPRISM GRAPHICS

The basic requirements of any range-angle map would be to 1) incorporate range and angle location information for each element within the data array, 2) provide polar axes and labeling, 3) provide a colorbar (or colorkey) label, and 4) provide a color scheme (or colormap) selection.

In order to use the existing XPRISM graphics within RLSTAP as the potential range-angle plotting module, it was first noted that the translation of the 2D range-angle map data from the implied rectangular to the polar co-ordinate systems could be accomplished by using the location data field provided in the VIFF image data format. The location data field passes X-Y co-ordinates to XPRISM for locating the image in X-Y-Z, or 3 dimensional space, where the Z co-ordinate is provided in the imagedata field.

If the location data field is not used, then the X-Y location is implied by the indices of the image data array, as previously implemented in modules using XPRISM.

It was also noted that any colorbar or colorkey associated with the plot could be created from another XPRISM execution. Thus two plots would have to be drawn. This is similar to the implementation under PVWAVE. However there would be no control over the placement or size of the colorbar graphics window, which could cause confusion when executing multiple XPRISM modules for which there are associated colorbars.

Also, XPRISM is only capable of providing rectangular co-ordinate axes for the plot. In order to obtain the desired polar axes, a second set of data defining the axes must be generated and overlaid on the plot. To label the polar axes, the annotate function must be used. The undesirable quality here is that although the polar axes will rescale when the graphics window is enlarged, the annotations are referenced by exact pixel location and will not move correspondingly. However, the user could move them manually or delete them if so desired.

Also, with the desired viewing angle, namely looking straight down onto the X-Y plane from above (refer to Figure 5-1), it was noted that the labels for the rectangular axes provided by XPRISM, were typically rendered improperly. For example, the X-axis numerical labels would be overwritten by the X-axis title. Also, it was noted that the Z-axis label could not be "turned off" without also turning off the X and Y labels. Thus, these labels would also need to be generated using the annotate function and once again they would not move proportionately when the graphics window is enlarged.

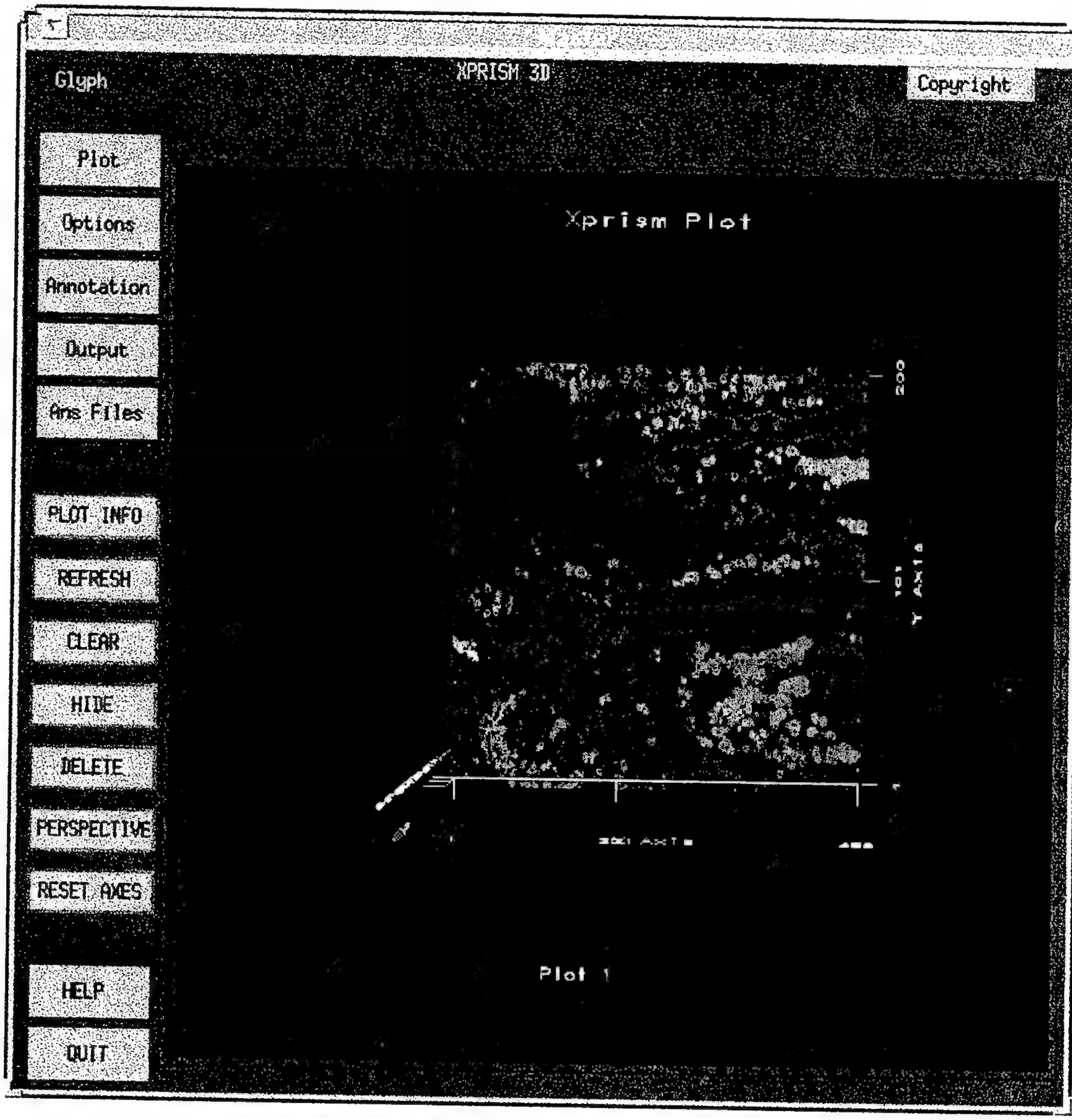

Figure 5-1. XPRISM3 Example with a 'Straight Down' View Angle

Finally, it was found that XPRISM did not allow for a separate colormap selection. Also, the colormap provided by XPRISM is not a good selection for the range-angle map as it starts with red, runs through the color spectrum back to red. Thus the highest and lowest values of the colorbar are red. When viewing the range-angle map from directly above, the highest and lowest values of the display are not unique. Also, the only method to provide a different colormap to XPRISM would be to significantly modify the Khoros code.

As it would be a copyright infringement to modify the XPRISM code, and with the colormap selection capability there would be other undesirable plot characteristics, as noted above, it was determined that XPRISM was insufficient and an Xlib graphics package was developed to provide this custom plot.

In developing a custom plot, several desirable features could be included: 1) the polar axes and labels could be drawn to expand appropriately when the graphics window was enlarged, 2) the colorbar could be incorporated into the same graphics window, 3) the rectangular axes labels would be rendered properly, and 4) the colormap selection could be provided. Also, other desirable characteristics could be incorporated, such as an interactive selection for the polar axes, scene plan, and colormap, etc. Thus, the "look and feel" of this custom plotting routine could be made to be more impressive and professional.

5.2 NEW XLIB IMPLEMENTATION

In order to insure that any user using RLSTAP could also use the range-angle map, only the Xlib and Athena widgets toolset was used to develop the graphics routine. Without these libraries, Khoros itself cannot be executed. Thus, any user capable of running Khoros will have these libraries available to them. Also, these libraries are available on several different platforms.

5.2.1 <u>Range Angle Maps</u>

The range-angle map plotting routine has been developed in a generic manner, so as to easily allow for future desired features. The software consists of the following code:

stap/include/range_angle_map.h

stap/include/range_angle_map.icon

stap/src/Lib/plotlib/range_angle_map.c

stap/src/Lib/plotlib/color_map_defs.c

This code is the backbone of any modules providing a range-angle map plot. These modules include RAMap, PltRAMap, and RafOzPlt. The module RAMap was the first module developed. It plots only the physical model simulation plots. The module PltRAMap is similar to RAMap, except that it also plots the measured CNR and Backscatter Coefficient maps. It also has the added feature of allowing the user to set the quadrant co-ordinates through it's control pane, thus providing a non-interactive zoom capability. The module RafOzPlt extracts the desired Ozturk statistic and plots this statistic in a range-angle map format. The graphic plotting routine itself, for all three modules is the same and is performed through a call to range_angle_plot.

The intent of this code was to provide only a basic functionality of the desired range-angle map. Future work should include adding more features to the range-angle map. However, at present this code satisfies all the basic requirements of a range-angle map plotting routine, as well as providing some additional features.

Figure 5-2 shows the graphics window generated by a call to the range-angle map plotting routine for the RA Terrain Height Map. Current features include the following interactive selections from the options bar located below the graphics image:

1. Interactive selection of 14 different colormaps.
2. Interactive "on/off" selection of polar axes.
3. Interactive "on/off" selection of scene plan.
4. Interactive "on/off" selection of image data.
5. Interactive selection of colorbar values with autoscale.
6. Interactive choice of reverse video display.
7. Interactive choice of the color of data <= zmin.

Other non-interactive features include the ability to 1) set the desired quadrant coordinates, thus providing a zoom capability, 2) provide alpha-numeric or numeric colorbar labels, and 3) preset the title and axes labels and all other plot characteristics.

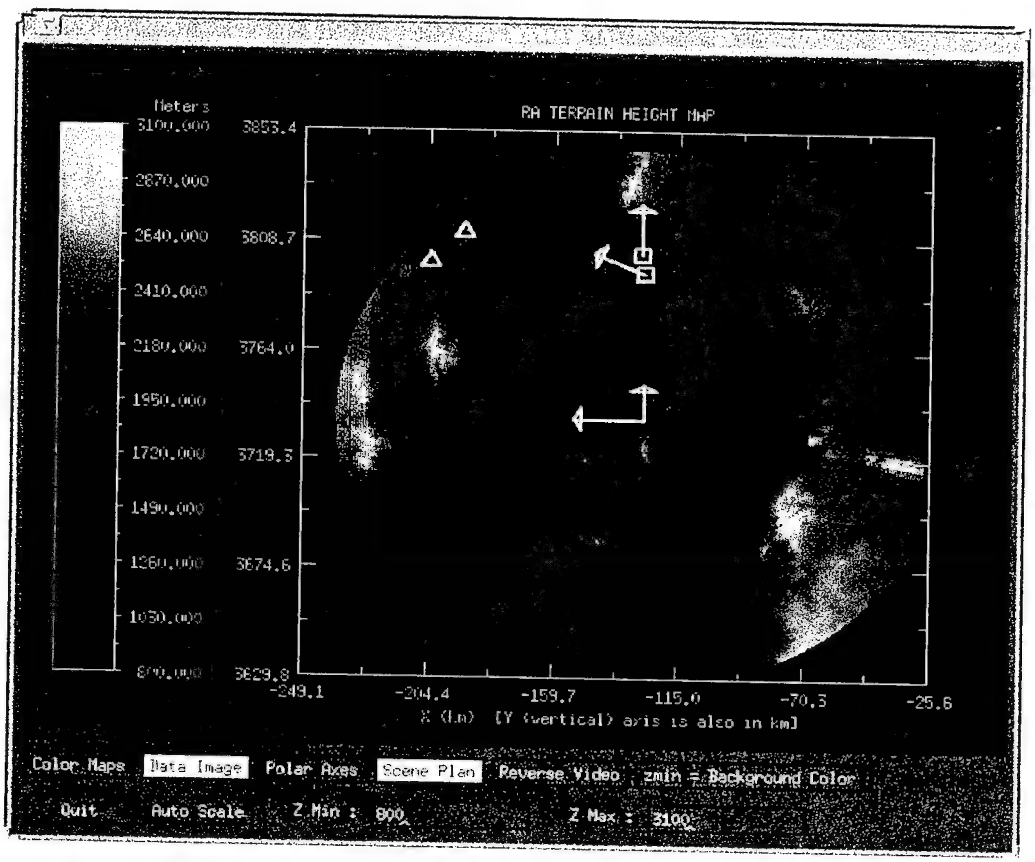

Figure 5-2. RAMap Example; RA Terrain Height Map with Scene Plan Overlay

Future features should include:

1. Interactive zoom capability.
2. Interactive cursor for raw source data retrieval.
3. Interactive annotation.

4. User defined title and axis labels.
5. Elevation contour map overlay.
6. Isodop contour map overlap.
7. Bilinear smoothing/interpolation.
8. Target, Jammer, Receiver trajectory overlay.

5.2.2 <u>XY Terrain Cover and Height Maps</u>

Previously the XY Terrain Cover and Height maps available within the physical model simulation were also provided by PVWAVE. For the same reason that XPRISM was insufficient for the range-angle maps, it is also insufficient for these maps as they also require a different colormap and require the straight down viewing angle. Thus, the module PltXYMap has been developed using the Xlib and Athena widgets toolsets to provide these maps. Figure 5-3 shows the resultant graphics window for the XY Terrain Height Map.

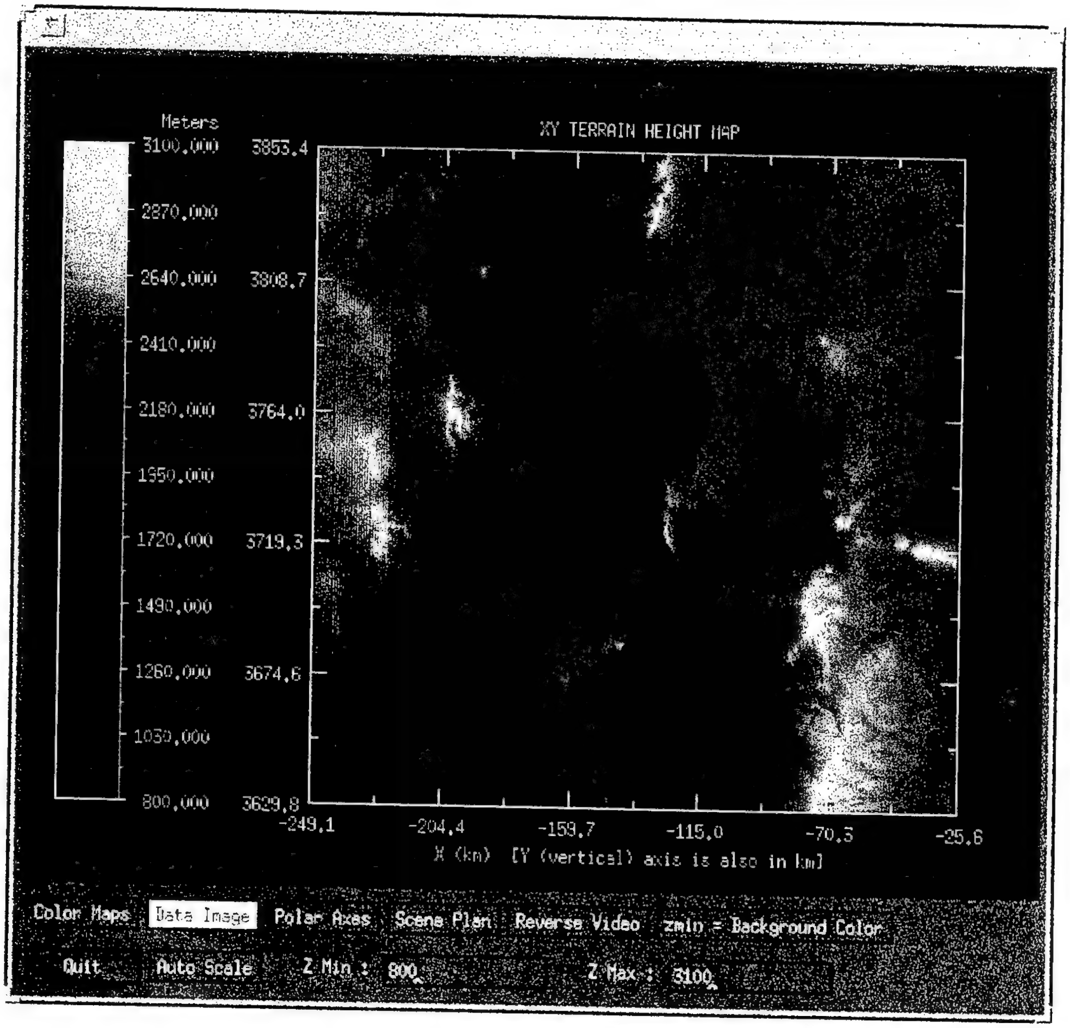

Figure 5-3. PltXYMap Example; XY Terrain Height Map

In this case, the basic code for the range-angle map was modified for the specific case of the XY Terrain Cover and Height maps. The major difference between these XY maps and the range-angle maps is that the USGS data file must be read in to produce the XY maps. This read is performed within the basic graphics code to provide as detailed a plot as possible for the desired quadrant.

Both the interactive and non-interactive features for the XY plots are the same as that for the range-angle plots.

6. IMPLEMENTATION OF THE OZTURK ALGORITHM LOOP APPLICATION

In previous sections we have discussed all the basic components of the Ozturk Algorithm loop application, particularly, 1) the method for acquiring measured clutter data, 2) the Khoros flow control modules used within the Ozturk Algorithm loop application, 3) the basic Ozturk Algorithm modules and 4) the range-angle map plotting routine. Thus, it is now time to put everything together to provide the automation of the Ozturk Algorithm.

Figure 6-1 provides the module lineup for the automated Ozturk Algorithm loop application. The input file of module RafSec was created in the lineup shown in Figure 1. This file consists of one range ring, defined from 0 to 360 degrees, in 1.8 degree increments and 2 frequencies (438 and 485 MHz) and has been generated by the method previously discussed.

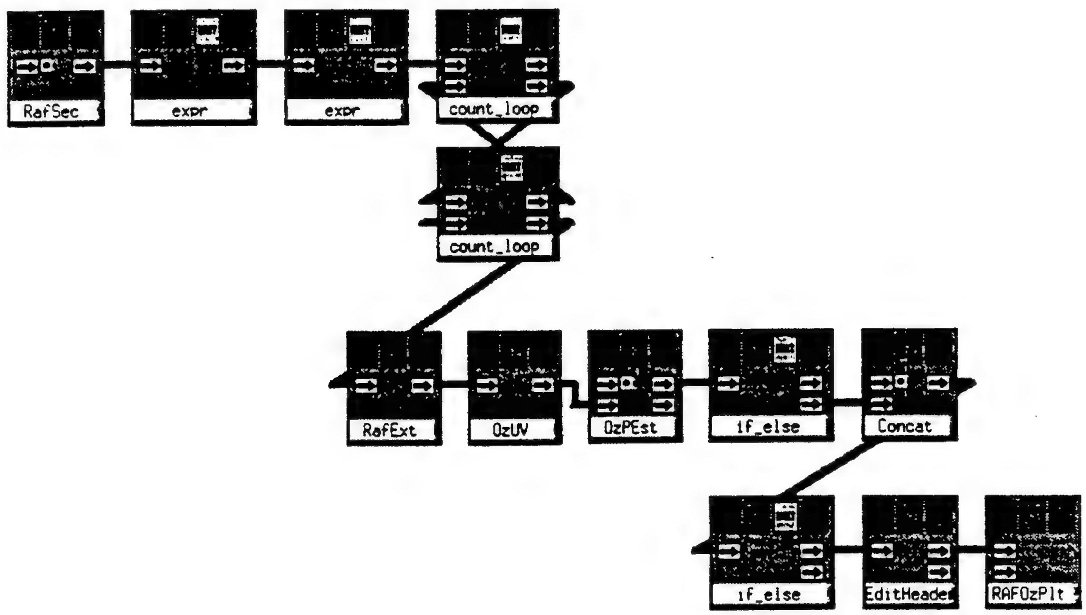

Figure 6-1. The Automated Ozturk Algorithm Loop Application Module Lineup

The control pane for RafSec is provided in Figure 6-2. This module sets up a data sector which will be slid in range and angle during the execution of the loop application. The definition describes how the module RafExt will extract the desired data. In this case, 25 points will be taken in range, 2 points in angle and 2 frequency points. RafExt will start each loop extraction at the current loop instances of range and angle and the first frequency point. Also from RafSec the points are to be taken at every 5th point in range and at every 4th point in angle. This is required for this data set, to obtain samples that are one resolution range cell apart and at least one azimuth beamwidth apart. The output of RafSec is simply the input data with a modified header to provide the desired definition to the module RafExt.

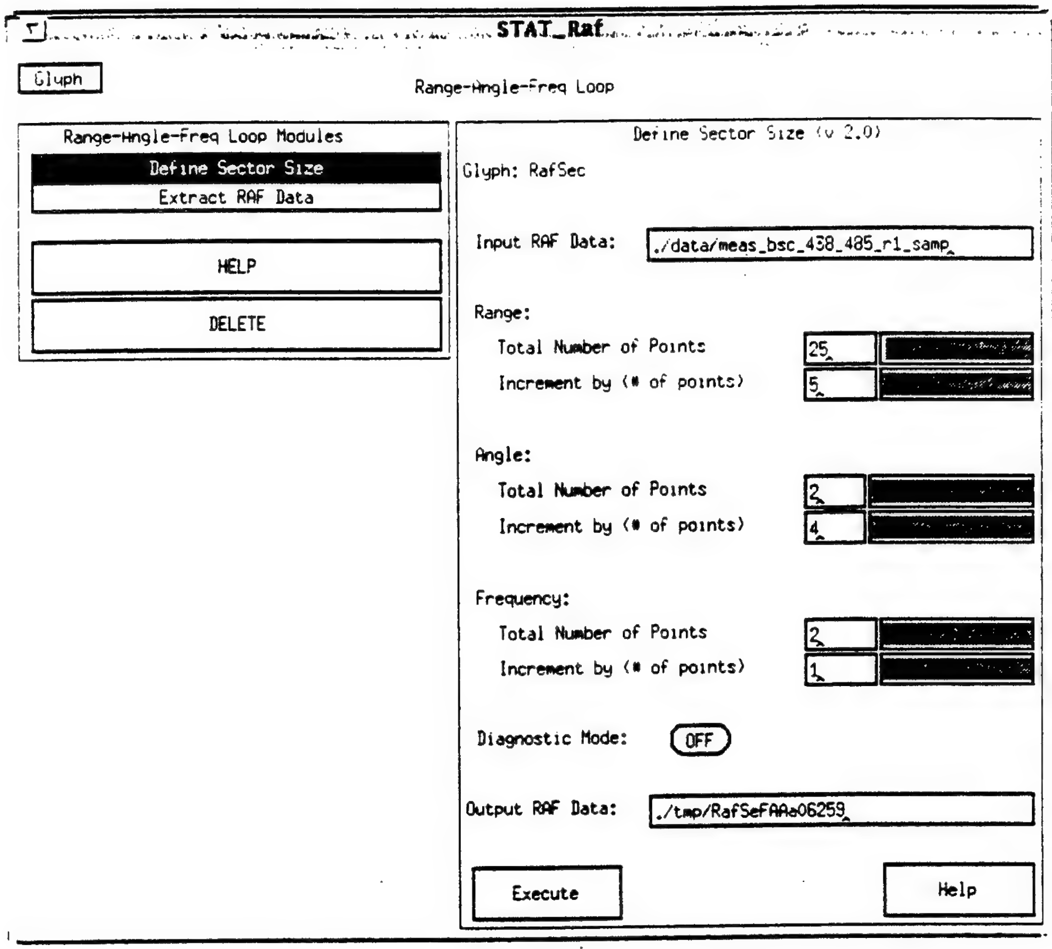

Figure 6-2. RafSec Module Control Pane

The control panes for the two expression modules (expr) are shown in Figures 6-3 and 6-4. These expressions are used as input arguments in the count_loop modules and the RafExt module. They define by index, the range and angle start, increment and end values for the range-angle-frequency (RAF) loop. The expression variables are placed within the count_loop and RafExt modules by including a "$" prefix to the variable name. RLSTAP then automatically provides the correct values to the control panes for these variables.

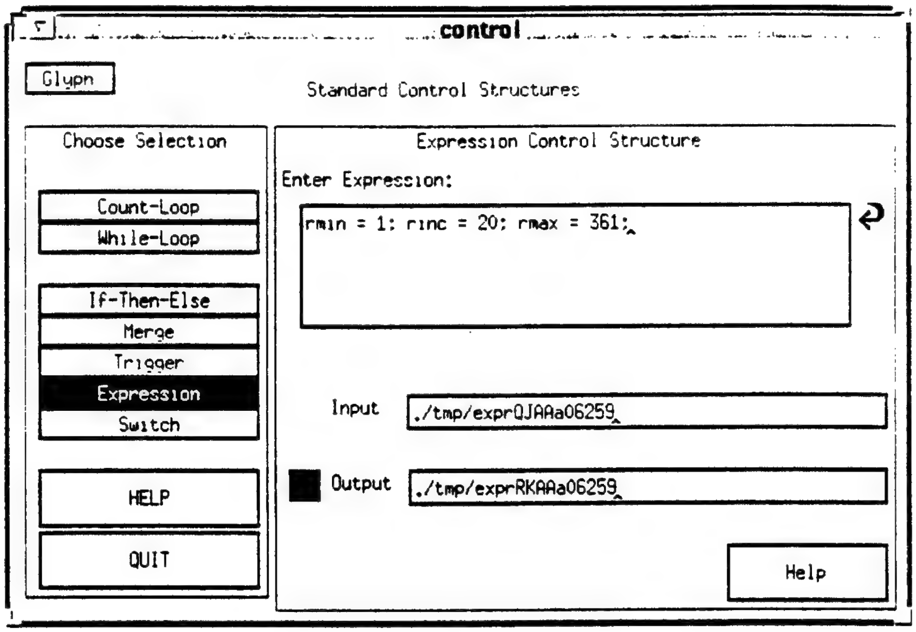

Figure 6-3. First Expression Module Control Pane

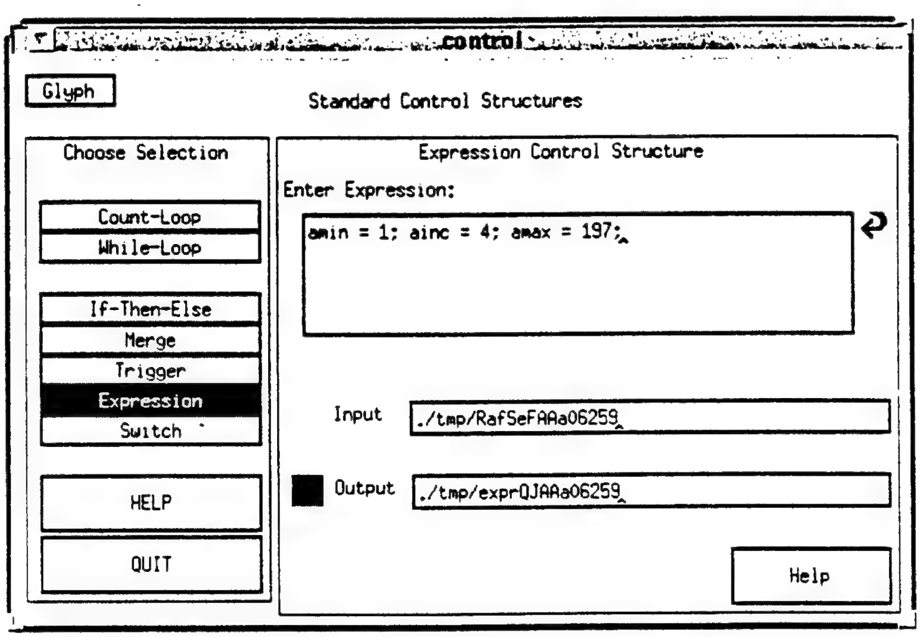

Figure 6-4. Second Expression Module Control Pane

The count_loop module control panes are provided in Figures 6-5 and 6-6. The inputs are simply as described above. The first count_loop is setup as a feedback loop, but the second is a no-feedback loop. For any no-feedback loop, the loop output is connected directly to the loop input as well as the desired modules within the loop. Note that the outer loop is the angle variable and the inner loop is the range variable. This is the necessary (assumed) setup which the module RafOzPlt will use to reshape the data.

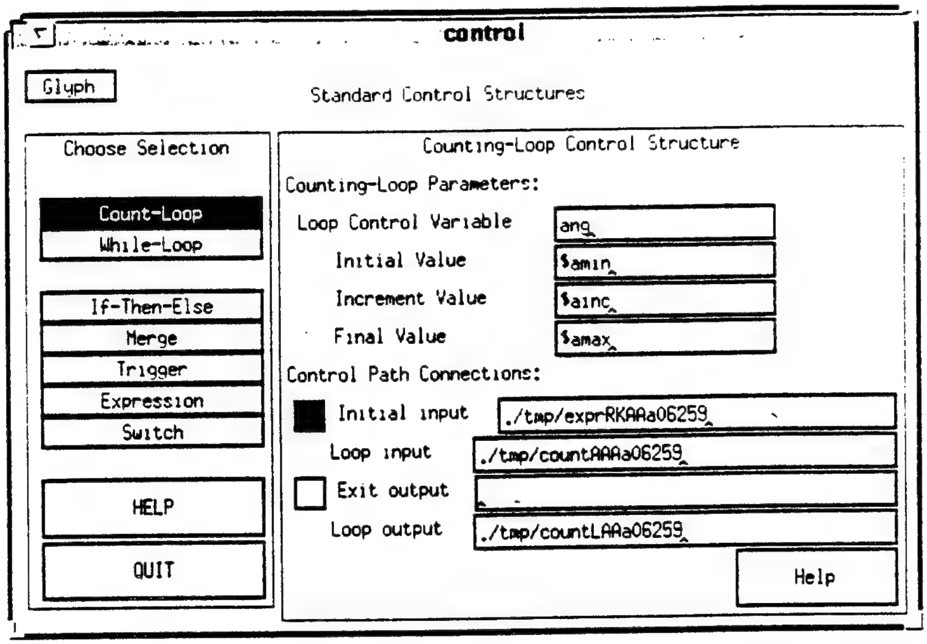

Figure 6-5. First (outer) Count_Loop Module Control Pane

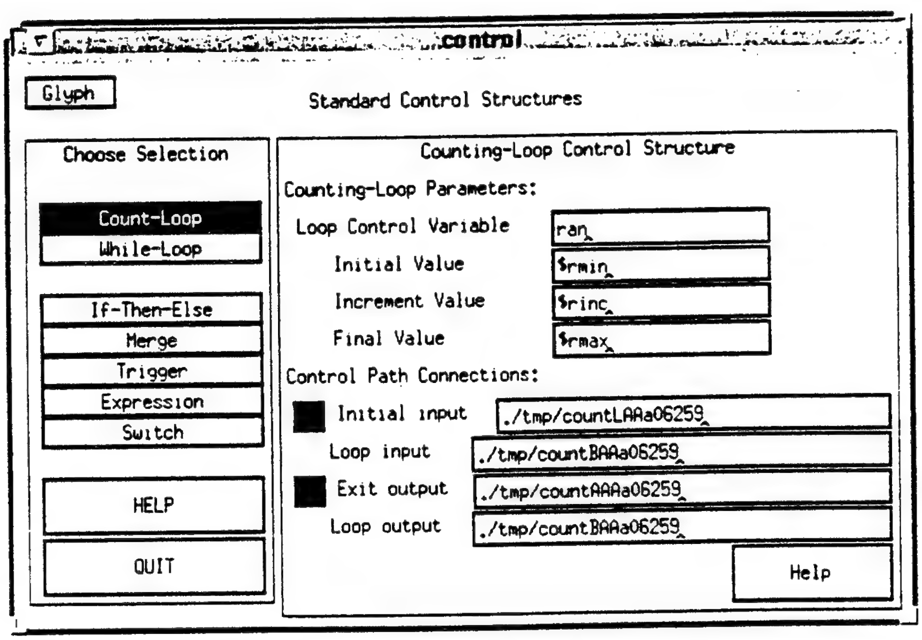

Figure 6-6. Second (inner) Count_Loop Module Control Pane

Thus, the input to the module RafExt will remain constant for each iteration of the loop. The desired input data to RafExt is the full range-angle-frequency data file that was provided as an input to RafSec, with the header modification made by RafSec.

The control pane for the module RafExt is shown in Figure 6-7. Note that the setup for the count_loop modules are repeated here. This is necessary since there is no

method to incorporate the RLSTAP VIFF header info into the Khoros flow control modules and the loop parameters that were setup need to be included in the VIFF header.

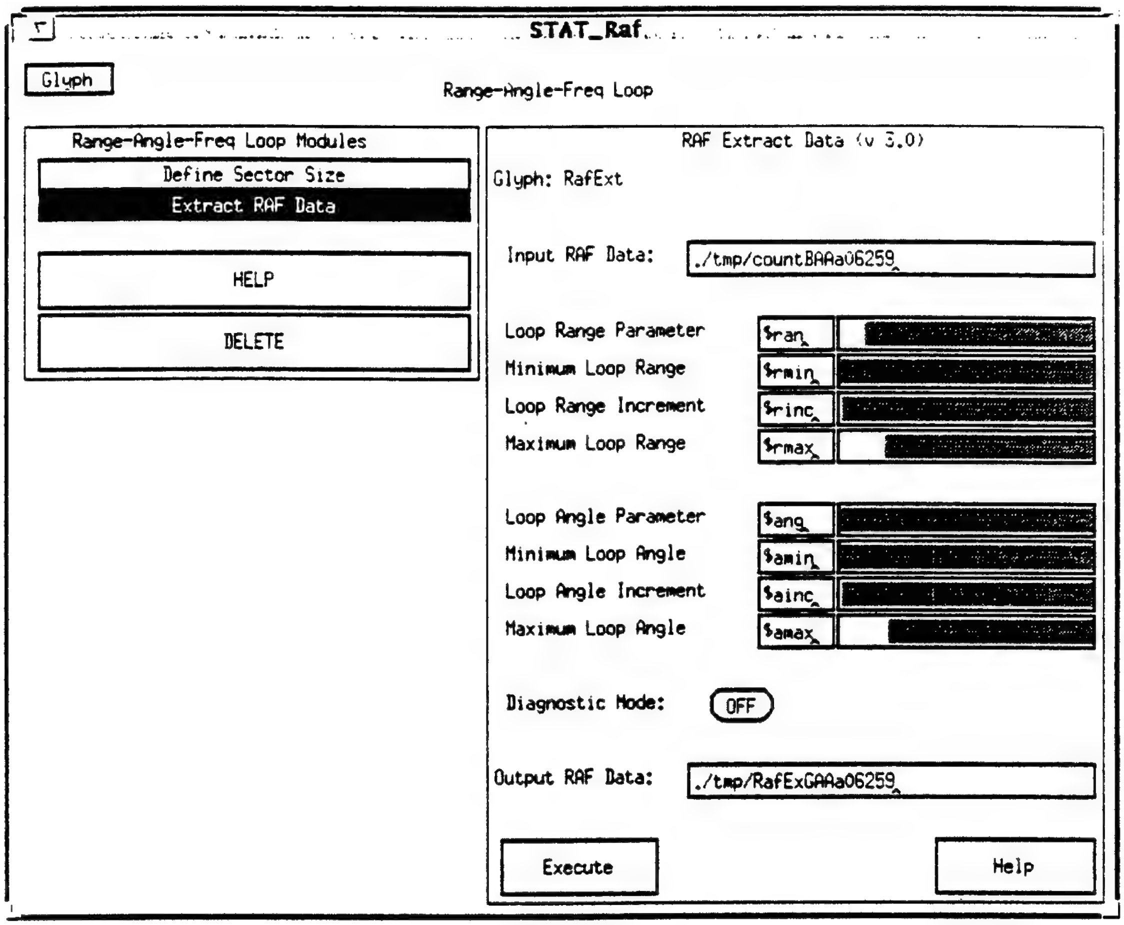

Figure 6-7. RafExt Module Control Pane

The RafExt module provides the loop control parameters to the VIFF header file on the first loop iteration (when $ang == $amin and $ran == $rmin). It then extracts the desired data based on the RafSec definition and the current loop parameters $ran and $ang. The range-angle-frequency data sector begins at the range and angle current value and the first frequency data point in the data array. The desired range-angle-frequency data array is then provided at the output.

The module OzUV then takes the range-angle-frequency data and determines the sample linked vector of the Ozturk statistic and provides it to the output.

The module OzPest then takes the sample data linked vector and the Ozturk approximation chart (both are of N=100 for this example) and determines the PDF parameter estimate. If the user defines the desired PDF to test against, say Weibull, then the output of OzPest will be an array of 8 Ozturk statistics (row), by 1 PDF (col).

The if_else module determines the occurrence of the first loop iteration and stores the data into a VIFF file, say "test". On subsequent iterations it passes the output to another filename through the second output. Thus, initializing the desired output filename each time the loop application is reset.

The Concat module takes the file "test" as the first input and any subsequent file generated for all other iterations and concatenates the data by bands and outputs it back into the file "test". Thus, it's output file will be a concatenation of all data: 8 Ozturk parameters (row), by 1 PDF (col), by the number of cases executed within the loop (band).

If the user had selected "calculate distribution" in the OzPest module, then the output data of OzPest would be 8 Ozturk statistics (row), by 16 rank implied PDFs (col). Thus the output of the Concat module would be 8 Ozturk statistics (row), by 16 rank implied PDFs (col), by the number of cases executed within the loop (band).

The second if_else module determines when the last loop iteration has occurred and starts the execution of the EditHeader and RafOzPlt modules at this time.

The EditHeader module is required at present due to the problem of the additional VIFF header file lines as discussed under section 3.2. These additional lines are placed in the VIFF header for each loop iteration. As RafOzPlt will be reading the VIFF header to find data towards the beginning of the file, it is necessary to delete some of the header information to speed up the read process. In some cases, a memory fault may occur, if this header file is too large (for a large number of iterations). An SPR has been turned in to address this issue.

During the execution of the EditHeader module, simply search for "ozuv" and delete from the first instance of this line to the end of the file, as none of this portion of the header file is required by RafOzPlt.

The module RafOzPlt allows the user to extract any of the 8 Ozturk statistics for a given rank PDF. In other words, if a given PDF was chosen by the user within OzPest, then RafOzPlt could extract any desired parameter from the chosen distribution. However, if "calculate distribution" was chosen for OzPest, then the desired parameter would be that associated with the PDF of a specific rank (i.e., rank = 1). The most useful situation for the automated Ozturk loop application is the first case.

If the Weibull distribution is chosen to test against, within the OzPest control pane, for each and every instance within the loop application, then by analyzing it's shape parameter, some insight may be gained as to how Gaussian the range-angle-frequency data might be. Since radar data that is Gaussian in the quadrature components is Rayleigh in magnitude and since the Weibull distribution is identically equal to the Rayleigh distribution for a shape parameter of 2.0, then investigating the shape parameter estimate of the Weibull distribution provides information as to the Gaussian nature of the radar clutter data.

However, even though the shape parameter estimate may be close to 2.0 for the Weibull case, does not guarantee that the data will be Gaussian in the quadrature components. This is due to the fact that the Ozturk statistic (U and V) must also fall within the confidence contours of the Goodness-of-Fit Test to insure that the data is statistically consistent with the Null Hypothesis.

Figure 6-8 provides the output of an automated Ozturk loop application with the input and parameters as described above. Note that these results are similar to that provided in Reference 1. Also, note that at present the range-angle map display under RLSTAP doesn't have an interpolation/smoothing function which was used for the graphics in Reference [1].

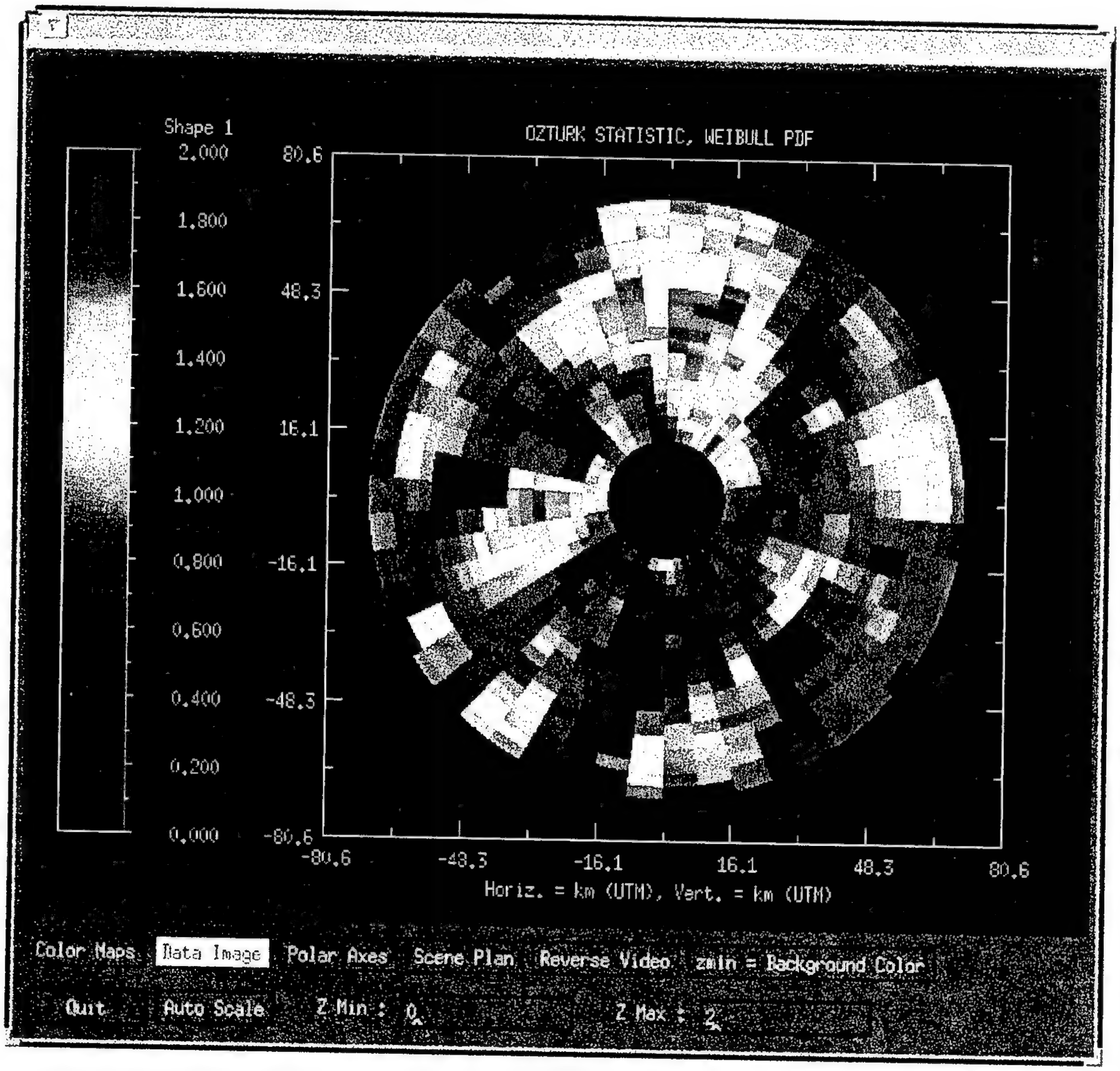

Figure 6-8. Automated Ozturk Algorithm Loop Application Results

7. CONCLUSION

The automated range-angle Ozturk Algorithm loop application was completed under RLSTAP and all necessary modules and change sets were provided for integration into RL's resident version. This effort included:

1) The development of software graphics modules dependent solely on the Xlib libraries on which RLSTAP is based to provide portable user graphics for plotting range-angle maps and XY terrain cover and height maps.

2) The development of software modules to read in measured clutter data from the RSTER radar.

3) The investigation of the implementation of the Khoros flow control modules to provide a loop mechanism for the automated Ozturk Algorithm loop application.

4) The validation and modification of the existing Ozturk Algorithm modules for use in a Khoros flow control loop.

5) The development of necessary modules to simplify the loop mechanism for the automated Ozturk Algorithm loop application.

6) The investigation and implementation of the automated Ozturk Algorithm loop application using the Khoros flow control modules and the modules developed in this effort to provide the loop mechanism.

In order to accomplish this effort, the researcher successfully communicated and co-ordinated with RL and other contractors associated with the RLSTAP code, primarily through the internet and telephone links. All software change sets were provided through the internet facilities to RL. Updates to the researcher's version of RLSTAP/ADT were obtained both through internet connections and tapes mailed from RL.

Finally, note that in order to not duplicate work performed by others, the researcher did not provide a histogram or PDF envelope module for RLSTAP as this was provided during this time frame by another contractor.

8. REFERENCES:

[1] L. Slaski, J. Maher, "PDF Approximation for Radar Data," Rome Lab Technical Report, RL-TR-95-32, April 1995.

[2] L. Slaski, Dr. M. Rangaswamy, "A New Efficient Algorithm for PDF Approximation," Rome Lab Technical Report, RL-TR-xx-xx.

[2] M. Rangaswamy, P. Chakravarthi, Dr. D. Weiner, Dr. L. Cai, Dr. H. Wang, Dr. A. Ozturk, "Signal Detection in Correlated Gaussian and Non-Gaussian Radar Clutter," Rome Lab Technical Report, RL-TR-93-79, Chapt. 6, 1993.

[3] A. Ozturk, E. J. Dudewicz, "A New Statistical Goodness-of-Fit Test Based on Graphical Representation," Technical Report no. 52, Department of Mathematics, Syracuse University, Syracuse, NY, 1990.

[4] A. Ozturk, "A General Algorithm for Univariate and Multivariate Goodness-of-Fit Tests Based on Graphical Representation," Communications in Statistics: Theory and Methods, vol. 20, no. 10, 1991.

[5] A. Ozturk, J. Romeu, "A new Method ofr Assessing Multivariate Normality with Graphical Applications," Communications in Statistics: Theory and Methods, vol. 21, no. 1, pp. 15-34, 1991.

[6] R. R. Shah, "A New Technique for Distribution Approximation of Random Data," Master's Thesis, Syracuse University, December 1993.

[7] J. Jayne, "Two-Dimensional Clutter Maps", Lincoln Laboratory Internal Memo, 23 Sep 1993.

[8] L. Slaski, "Creating Loops within the RLSTAP/ADT", memo provided to RL under contract F30602-93-D-0081, Task 8, 25 Oct 95.

9. APPENDIX A. MODULE DEVELOPMENT

Many modules were required to be developed or modified as part of this effort. These modules were provided for integration into the version of the RLSTAP/ADT testbed resident at Rome Lab. For every module, a test procedure and Interface Control Document (ICD) was also provided as part of a software change set.

9.1 LIST OF MODULES PROVIDED

To accomplish the effort the following modules were validated, extensively modified or developed:

To read in range-angle RSTER radar clutter data:

RSTERcnr: (Developed) This module reads in the appropriate data files created by RSTERin, performs a beamform operation across the channels, divides the result by the noise estimate, and outputs a range-angle data structure with explicit location for the CNR data.

RSTERbsc: (Developed) This module reads in the CNR data and radar parameters and outputs a range-angle data structure with explicit location for the Backscatter Coefficient data.

RSTERmsh: (Developed) This module reads in up to three multiple overlapping range rings of range-angle map data from RSTERcnr or RSTERbsc and outputs a single range-angle data map of sampled data with implicit location.

To apply the Ozturk Algorithm:

OzID: (Validated) This module obtains the Ozturk Approximation (or Identification) Chart for a given sample set size.

OzGen: (Validated) This module provides random samples from any of the 16 PDF's available within the Ozturk Algorithm.

OzGof: (Validated) This module computes the Goodness-of-Fit test and resultant graphics for the Ozturk Algorithm.

OzUV: (Modified) This module determines the Ozturk sample data linked vector in the [U,V] co-ordinate system.

OzPest: (Modified) This module estimates the PDF shape parameters.

OzLoc: (Modified) This module estimates the PDF location and scale parameters.

OzGraph: (Modified) This module provides the display of the graphical result of the Ozturk Algorithm PDF approximation.

To apply the Ozturk Algorithm using the Khoros flow controls:

RAFSec: (Developed) This module defines the desired range-angle-frequency sector which will be allowed to move within the loop.

RAFExt: (Developed) This module extracts the range-angle-frequency data from the desired sector for each loop instance.

RAFOzPlt: (Developed) This module extracts the desired Ozturk statistic from the final loop result and displays it in a range-angle map format.

Other diagnostic tools developed:

RAMap: (Developed) This module uses the Xlib library routines to replace the range-angle map plotting modules dependent on PVWAVE. The module displays a range-angle map for the physical model simulations.

PltRAMap: (Developed) This module is a modified version of RAMap to display the measured clutter range-angle maps as well as the physical model simulations.

PltXYMap: (Developed) This module was developed to use the Xlib library routines to replace the XY Terrain Cover and Height modules dependent on PVWAVE.

9.2 LIST OF SOFTWARE CHANGE SETS PROVIDED

The following is a list of the software change sets provided to RL with a basic description of each change set.

SPR_142, SPR_142.1, SPR_142.2

This is the first change set provided for the range-angle map plotting utility RAMap.

Modules provided: RAMap

SPR_246

This change set split the module CNRMap into CNRCalc which computes the values of the CNR map data and CNRMap which performs only the PVWAVE graphics to obtain a display utility. It also created a new submenu for the PVWAVE routines and all these routines were moved to this submenu. This change set has yet to be implemented.

Modules provided: CNRCalc

Modules modified: CNRMap

SPR_239

This change set provides all of the basic Ozturk Algorithm modules necessary to process a given sample data set.

Modules provided: OzUV, OzLoc, OzPest, OzGraph

Modules modified: OzID, OzGOF, OzGen

SPR_240

This change set provides all of the necessary range-angle-frequency loop application specific modules to perform the Ozturk Algorithm within a Khoros flow control count-loop. It also provided the module necessary to compute the Backscatter Coefficient from the RSTER radar data.

Modules provided: RAFExt, RAFSec, RAFOzPlt, RSTERbsc

SPR_261

This change set provided a correction to the labeling scheme for the module RAMap.

Modules modified: RAMap

SPR_307

This change set provided the modules required to read in the RSTER radar range-angle clutter data and to plot this data.

Modules provided: RSTERcnr, RSTERmsh, PltRAMap

SPR_352

This change set provided the module required to plot the XY Terrain Cover and Height Maps from the physical model simulation, using the Xlib libraries instead of PVWAVE.

Modules provided: PltXYMap

SPR_362

This change set provided an update to the RAMap and PltRAMap modules to provide the user with an option to plot the graphics such that the receiver is located at the center of the image window.

Modules modified: RAMap, PltRAMap

DISTRIBUTION LIST

addresses	number of copies
ROME LABORATORY/OCSA ATTN: E. DOUGLAS LYNCH 26 ELECTRONIC PKY ROME, NY 13441-4514	4
DECISION-SCIENCE APPLICATIONS, INC. 1300 FLOYD AVE., BLDG B ROME, NY 13440	2
ROME LABORATORY/SUL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ADVANCED RESEARCH PROJECTS AGENCY STO ATTN: MAJ. DAN GAMMON 3701 N. FAIRFAX DRIVE ARLINGTON, VA 22203-1714	1
SAIC ATTN: MR. TRAVIS SLOCUMB 4001 N. FAIRFAX DRIVE SUITE 500 ARLINGTON, VA 22203	1
MS. LISA SLASKI 13325 MALLARD COVE BLVD ORLANDO, FL 32837	4

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.